Exact Sampling of Determinantal Point Processes without Eigendecomposition

Claire Launay, Bruno Galerne, Agnès Desolneux

Exact Sampling of Determinantal Point Processes without Eigendecomposition

Claire Launay[1], Bruno Galerne[1], and Agnès Desolneux[2]

[1] *Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité*
[2] *CMLA, ENS Cachan, CNRS, Université Paris-Saclay*

Abstract

Determinantal point processes (DPPs) enable the modelling of repulsion: they provide diverse sets of points. This repulsion is encoded in a kernel K that we can see as a matrix storing the similarity between points. The usual algorithm to sample DPPs is exact but it uses the spectral decomposition of K, a computation that becomes costly when dealing with a high number of points. Here, we present an alternative exact algorithm that avoids the eigenvalues and the eigenvectors computation and that is, for some applications, faster than the original algorithm.

1 Introduction

Determinantal point processes (DPPs) are processes that capture negative correlations. The more similar two points are, the less likely they are to be sampled simultaneously. Then, DPPs tend to create sets of diverse points. They naturally arise in random matrix theory [7]. Ever since [12], these processes have become more and more popular in machine learning, thanks to their ability to draw subsamples that account for the inner diversity of the datasets. This property is useful for many applications, such as summarizing documents [3], generating diverse subsamples of datasets to improve a stochastic gradient descent [20], selecting appropriate batches of experiments to fasten a Bayesian optimisation problem [10] or modelling a natural repulsive phenomenon like the repartition of trees in a forest [14]. Several issues are under study, as learning DPPs, for instance through maximum likelihood estimation [13], or sampling these processes. Here we will focus on the sampling question and we will only deal with discrete and finite determinantal point processes, particularly adapted to machine learning groundsets. A determinantal process Y is defined by its matrix K and the repulsion comes from the fact that the probabilities of inclusion of the process are related to the determinant of K.

The main algorithm to sample DPPs is a spectral algorithm [8] : it uses the eigendecomposition of K to sample Y. It is exact and in general quite fast. Yet, the computation of the eigenvalues of K may be very costly when dealing with large-scale data. That is why numerous algorithms has been conceived to bypass this issue. Some authors tried to design a sampling algorithm adapted to specific DPPs. For instance, it is possible to speed the initial algorithm up by assuming that K has a bounded rank [11, 4]. Then these authors use a dual representation so that almost all the computations in the spectral algorithm are reduced. One can also deal with another class of DPPs associated to kernels K that can be decomposed in a sum of tractable matrices [3]. In this case, the sampling is much faster and they study the inference on these classes of DPPs. At last, Propp and Wilson [17] use Markov chains and the theory of coupling from the past to sample exactly particular DPPs : uniform spanning trees. Another type of sampling algorithms is the class of approximate methods. Some authors approach the original DPP with a low rank matrix to be able to apply the previous dual representation, either by random projections [12, 6] or thanks to the Nystrom approximation [1]. The Markov Chain Monte Carlo methods offer also nice approximate sampling algorithms for DPPs. It is possible to obtain satisfying convergence guarantees for particular DPPs (for instance, k-DPPs with fixed cardinal [2, 15] or projection DPPs [5]), and even a polynomial-time sampling algorithm for general DPPs [16], thus correcting the initial work of [9].

As one can see, except the initial spectral algorithm, no algorithm allows for the exact sampling of a generic DPP. The main contribution of this paper is to introduce such a general algorithm that does not involve kernel eigndecomposition. The proposed algorithm is a sequential thinning procedure that relies on two new results: (i) the explicit formulation of the marginals of any determinantal point process and (ii) the derivation of an adapted Bernoulli point process containing a given DPP.

The rest of the paper is organized as follows : in

the next section, we present the general framework of determinantal point processes and the classic spectral algorithm. In Section 3, we provide an explicit formulation of the general marginals and pointwise conditional probabilities of any determinantal point process, from its kernel K. Thanks to these formulations, we develop a "naive", exact but slow, sequential algorithm. In Section 4, using the sequential thinning theory, we introduce a new exact sampling algorithm for DPPs and in Section 5, our experiments show that for some applications, this algorithm is even faster than the spectral algorithm. Finally, we discuss and conclude around this algorithm.

2 DPPs and their Usual Sampling Method

In all the paper, we will use the following notations. Let's consider a discrete finite set $\mathcal{Y} = \{1, \dots, N\}$. We will denote by $M_{A\times B}$, $\forall A, B \subset \mathcal{Y}$, the matrix $(M(i,j))_{(i,j)\in A\times B}$ and the short notation $M_A = M_{A\times A}$. Suppose that K is a Hermitian positive semi-definite matrix of size $N \times N$, indexed by the elements of $\mathcal{Y}$, so that any of its eigenvalues is in $[0, 1]$. A subset $Y \subset \mathcal{Y}$ is said to follow a DPP distribution of kernel K if, $\forall A \subset \mathcal{Y}$, $\mathbb{P}(A \subset Y) = \det(K_A)$.

The spectral algorithm 1 is standard to draw a DPP. It relies on the eigendecompostition of its kernel K. It was first introduced by [8] and is also presented in a more detailed way in [19, 12, 14]. It consists in first randomly selecting a set of active eigenvectors and then drawing sequentially the associated points.

This algorithm is exact and relatively fast. Nevertheless, when the size of $\mathcal{Y}$ grows, the matrix K does too and computing its eigendecomposition becomes heavy. We will see in the numerical results that this first step represents in general more than 90% of the running time of the spectral algorithm. We show below that any DPP can be exactly sampled by a concurrent algorithm that does not require the eigendecomposition of K.

3 Sequential Sampling Algorithm

3.1 Explicit General Marginal of a DPP

To develop our first "naive" sequential sampling algorithm, we need to explicit the marginals and the conditional probabilities of any DPP. We show below that they can easily be formulated from the associated kernel matrix K. $\forall A \subset \mathcal{Y}$, we denote I^A the

Algorithm 1 The spectral sampling algorithm

1. Eigendecomposition (λ_j, v^j) of the matrix K.
2. Selection of active frequencies: Draw a Bernoulli process $\mathbf{X} \in \{0,1\}^N$ with parameter $(\lambda_j)_j$.

   Denote by n the number of active frequencies, $\{\mathbf{X} = 1\} = \{j_1, \dots, j_n\}$. Define the matrix $V = \begin{pmatrix} v^{j_1} & v^{j_2} & \dots & v^{j_n} \end{pmatrix} \in \mathbb{R}^{N\times n}$ and denote by $V_k \in \mathbb{R}^n$ the k-th line of V, for $k \in \mathcal{Y}$.
3. Return the sequence $Y = \{y_1, y_2, \dots, y_n\}$ sequentially drawn as shown:
   For $l = 1$ to n
   - Sample a point $y_l \in \mathcal{Y}$ from the discrete distribution,

$$p_k^l = \frac{1}{n-l+1}\left(\|V_k\|^2 - \sum_{m=1}^{l-1} |\langle V_k, e_m\rangle|^2\right), \forall k \in \mathcal{Y}.$$

   - If $l < n$, define $e_l = \frac{w_l}{\|w_l\|} \in \mathbb{R}^n$ where $w_l = V_{y_l} - \sum_{m=1}^{l-1}\langle V_{y_l}, e_m\rangle e_m$.

$N \times N$matrix with 1 on its diagonal coefficients indexed by the elements of A, and 0 anywhere else. We also denote $\overline{A}$ the complementary set of A in $\mathcal{Y}$.

Proposition 3.1 (Distribution of a DPP)**.** *For $A \subset \mathcal{Y}$, $\mathbb{P}(Y = A) = (-1)^{|A|}\det(I^{\overline{A}} - K)$.*

Proof. We have that $\mathbb{P}(A \subset Y) = \sum_{B\supset A} \mathbb{P}(Y = B)$. Thanks to the Möbius inversion formula, for all $A \subset \mathcal{Y}$,

$$\begin{aligned}\mathbb{P}(Y = A) &= \sum_{B\supset A} (-1)^{|B\setminus A|}\mathbb{P}(B \subset Y)\\ &= (-1)^{|A|}\sum_{B\supset A}(-1)^{|B|}\det(K_B)\\ &= (-1)^{|A|}\sum_{B\supset A}\det((-K)_B)\\ &= (-1)^{|A|}\det(I^{\overline{A}} - K),\end{aligned}$$

where for the last step we used Theorem 2.1 of [12]. □

We have by definition $\mathbb{P}(A \subset Y) = \det(K_A)$ for all A, and as a consequence $\mathbb{P}(B \cap Y = \emptyset) = \det((I-K)_B)$ for all B. Moreover, when dealing with a L-ensemble L rather than with the kernel K, which is possible as soon as $I - K$ is invertible, one can formulate the explicit marginals of the DPP [12]. The next proposition gives for any DPP the expression of the general marginal $\mathbb{P}(A \subset Y, B \cap Y = \emptyset)$, for A, B disjoint subsets of $\mathcal{Y}$,

using K. In what follows, H^B denotes the symmetric positive semi-definite matrix

$$H^B = K + K_{\mathcal{Y}\times B}((I-K)_B)^{-1}K_{B\times\mathcal{Y}}.$$

Theorem 3.1 (General Marginal of a DPP)**.** *Let $A, B \subset \mathcal{Y}$ be disjoint. If $\mathbb{P}(B \cap Y = \emptyset) = \det((I-K)_B) = 0$, then $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = 0$. Otherwise, the matrix $(I-K)_B$ is invertible and*

$$\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I-K)_B)\det(H^B_A).$$

Proof. Let $A, B \subset \mathcal{Y}$ disjoint such that $\mathbb{P}(B \cap Y = \emptyset) \neq 0$. Using the previous proposition,

$$\begin{aligned}\mathbb{P}(A \subset Y, B \cap Y = \emptyset) &= \sum_{A\subset C\subset\overline{B}} \mathbb{P}(Y = C) \\ &= \sum_{A\subset C\subset\overline{B}} (-1)^{|C|}\det(I^{\overline{C}} - K).\end{aligned}$$

For any C such that $A \subset C \subset \overline{B}$, one has $B \subset \overline{C}$. Hence, by reordering the matrix coefficients, and using the Schur's determinant formula,

$$\begin{aligned}\det(I^{\overline{C}} - K) &= \det\begin{pmatrix}(I^{\overline{C}}-K)_B & (I^{\overline{C}}-K)_{B\times\overline{B}} \\ (I^{\overline{C}}-K)_{\overline{B}\times B} & (I^{\overline{C}}-K)_{\overline{B}}\end{pmatrix} \\ &= \det\begin{pmatrix}(I-K)_B & -K_{B\times\overline{B}} \\ -K_{\overline{B}\times B} & (I^{\overline{C}}-K)_{\overline{B}}\end{pmatrix} \\ &= \det((I-K)_B)\det((I^{\overline{C}} - H^B)_{\overline{B}}).\end{aligned}$$

$$\begin{aligned}&\text{Thus, } \mathbb{P}(A \subset Y, B \cap Y = \emptyset) \\ &= \det((I-K)_B)\sum_{A\subset C\subset\overline{B}} (-1)^{|C|}\det((I^{\overline{C}} - H^B)_{\overline{B}}).\end{aligned}$$

According to [12], for all $A \subset \overline{B}$,

$$\sum_{A\subset C\subset\overline{B}} \det(-H^B_C) = \det((I_{\overline{A}} - H^B)_{\overline{B}}).$$

Then, Möbius inversion formula ensures that, $\forall A \subset \overline{B}$,

$$\begin{aligned}\sum_{A\subset C\subset\overline{B}} (-1)^{|C\setminus A|}\det((I^{\overline{C}} - H^B)_{\overline{B}}) &= \det(-H^B_A) \\ &= (-1)^{|A|}\det(H^B_A).\end{aligned}$$

Hence, $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I-K)_B)\det(H^B_A)$. □

Thanks to this formula, we can explicitely formulate the pointwise conditional probabilities of any DPP.

Corollary 3.1 (Pointwise conditional probabilities of a DPP)**.** *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y,\ B \cap Y = \emptyset) \neq 0$, and let $k \notin A \cup B$. Then,*

$$\begin{aligned}\mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset) &= \frac{\det(H^B_{A\cup\{k\}})}{\det(H^B_A)} \\ &= H^B(k,k) - H^B_{\{k\}\times A}(H^B_A)^{-1}H^B_{A\times\{k\}}.\end{aligned} \quad (1)$$

This is a straightforward application of the previous expression and the Schur determinant formula. Now, we have all the necessary expressions for the sequential sampling of a DPP.

3.2 Sequential Sampling Algorithm of a DPP

This sequential sampling algorithm simply consists in using Formula (1) and updating at each step the pointwise conditionnal probability, knowing the previous selected points. It is presented in Algorithm 2.

Algorithm 2 Sequential sampling of a DPP with kernel K

- Initialization: $A \leftarrow \emptyset$, $B \leftarrow \emptyset$.
- For $k = 1$ to N:

1. Compute $H^B_{A\cup\{k\}}$
   $= K_{A\cup\{k\}} + K_{A\cup\{k\}\times B}((I-K)_B)^{-1}K_{B\times A\cup\{k\}}$.
2. Compute the probability p_k given by
$$\begin{aligned}p_k &= \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset) \\ &= H^B(k,k) - H^B_{\{k\}\times A}(H^B_A)^{-1}H^B_{A\times\{k\}}.\end{aligned}$$
3. With probability p_k, k is included, $A \leftarrow A \cup \{k\}$, otherwise $B \leftarrow B \cup \{k\}$.

- Return A.

The main operations of Algorithm 2 solve linear systems $(I-K)^{-1}_B$. Fortunately, here we can use the Cholesky factorization, which alleviates the computational cost. Suppose that L^B is the Cholesky factorization of $(I-K)_B$, that is, L^B is a lower triangular matrix such that $(I-K)_B = L^B(L^B)^*$ (where $(L^B)^*$ is the conjugate transpose of L^B). Then, denoting $J^B = (L^B)^{-1}K_{B\times A\cup\{k\}}$, one simply has $H^B_{A\cup\{k\}} = K_{A\cup\{k\}} + (J^B)^*J^B$.

Besides, at each iteration, the Cholesky decomposition $L^{B\cup\{k\}}$ of $(I-K)_{B\cup\{k\}}$ can be computed from L^B using standard Cholesky update operations, involving the resolution of only one linear system of size $|B|$.

In comparison with the spectral sampling algorithm of [8], one requires computations for each site of $\mathcal{Y}$,

and not just one for each sampled point of Y. We will see indeed that it is not really competitive. However, in what follows, we show that we can significantly decrease the number of steps and the running time of this first algorithm: we propose to simulate a point process X containing Y, the desired DPP, and then make a sequential selection of the points of X to obtain Y. This procedure can be called a sequential thinning.

4 Sequential Thinning Algorithm

4.1 General framework of sequential thinning

We attempt at describing a general sufficient condition for which a target point process Y - it will be a determinantal point process in our case - can be obtained as a sequential thinning of a point process X. This is a discrete adaptation of the thinning procedure on the continuous line of [18]. To do this, we will consider a coupling (X, Z) such that $Z \subset X$ is a random selection of the points of X and has the same distribution as Y. From this point onwards, we identify the set X with the vector of size N with 1 in the place of the elements of X and 0 elsewhere, and we use the notation $X_{1:k}$ to denote the vector $(X_1, \dots, X_k)$. We want to define the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N) \in \mathbb{R}^{2N}$ with the following conditional distributions for X_k and Z_k:

$$\begin{cases} \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ \qquad = \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}) \\ \\ \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k} = x_{1:k}) \\ \qquad = \mathbb{1}_{\{x_k = 1\}} \dfrac{\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1})}{\mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1})}. \end{cases} \tag{2}$$

Proposition 4.1 (Sequential thinning)**.** *Assume that X, Y, Z are discrete point processes on $\mathcal{Y}$ that satisfy for all $k \in \{1, \dots, N\}$, and all z, $x \in \{0,1\}^N$,*

$$\begin{aligned} &\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0 \\ &\qquad \textit{implies} \\ &\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \\ &\qquad \leq \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}). \end{aligned} \tag{3}$$

Then, it is possible to choose (X, Z) in such a way that (2) is satisfied. In that case, we moreover have that Z is a thinning of X ($Z \subset X$), and Z has the same distribution as Y.

Proof. Let us first discuss the definition of the coupling (X, Z). Thanks to the conditions (3), the ratios defining the conditional probabilities of Equation (2) are ensured to be between 0 and 1 (if the conditional events have non zero probabilities). Hence the conditional probabilities permits to construct sequentially the distribution of the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N)$ of length $2N$, and thus the coupling is well-defined. Besides, clearly, one has $Z \subset X$.

Let us now show that Z has the same distribution as Y. By complementarity of the events $\{Z_k = 0\}$ and $\{Z_k = 1\}$, it is enough to show that for all $k \in \{1, \dots, N\}$, and $z_1, \dots, z_{k-1}$ such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$,

$$\begin{aligned} &\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}) \\ &\quad = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}). \end{aligned} \tag{4}$$

Let $k \in \{1, \dots, N\}$, $(z_{1:k-1}, x_{1:k-1}) \in \{0,1\}^{2(k-1)}$, such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$. Since $Z \subset X$, $\{Z_k = 1\} = \{Z_k = 1, X_k = 1\}$. Suppose first that $\mathbb{P}(X_k = 1 | X_1 = x_1, \dots, X_{k-1} = x_{k-1}) \neq 0$.

$$\begin{aligned} &\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \mathbb{P}(Z_k = 1, X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \begin{aligned}[t] &\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}, X_k = 1) \\ &\times \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \end{aligned} \\ &= \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}), \text{ by Equations (2).} \end{aligned}$$

If $\mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}) = 0$, then $\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = 0$ and thanks to (3), $\mathbb{P}(Y_k = 1 | Y_{1:k} = z_{1:k}) = 0$. Hence the identity,

$$\begin{aligned} &\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &\quad = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \end{aligned}$$

is always valid. Since the values $x_1, \dots, x_{k-1}$ do not influence this conditional probability, one can conclude that given $(Z_1, \dots, Z_{k-1})$, Z_k is independent of $X_1, \dots, X_{k-1}$, and thus (4) is true. □

The characterization of the thinning defined here allows both extreme cases: there can be no pre-selection of points by X, meaning that $X = \mathcal{Y}$ and that the DPP Y is sampled by the usual sequential sampling algorithm, or there can be no thinning at all, meaning that the final process Y can be equal to the dominating process X. Regarding sampling acceleration, a good dominating process X must be sampled fastly and with a cardinal as close as possible to $|Y|$.

4.2 Sequential thinning algorithm for DPPs

In this section, we use the sequential thinning approach, where Y is a DPP of kernel K on the groundset $\mathcal{Y}$,

and X is a Bernoulli point process (BPP). BPPs are the fastest and easiest point processes to sample. X is a Bernoulli process if the components of the vector $(X_1, \dots, X_N)$ are independent. Its distribution is determined by the probability of occurrence of each point k, that we denote by $q_k = \mathbb{P}(X_k = 1)$. Thanks to the independence property the conditions (3) simplifies to

$$\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0 \text{ implies } \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \leq q_k.$$

The second inequality does not depend on x, hence it must be valid as soon as there exists a vector x such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$, that is, as soon as $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$. Since we want Z to have the same distribution as Y, we finally obtain the conditions

$$\forall y \in \{0,1\}^N, \quad \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0 \text{ implies } \mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}) \leq q_k.$$

Ideally, we want the q_k to be as small as possible to ensure that the cardinal of X is as small as possible. So we look for the optimal values q_k^*, that is,

$$q_k^* = \max_{\substack{(y_{1:k-1}) \in \{0,1\}^{k-1} \text{ s.t.} \\ \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0}} \mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}).$$

A priori, computing q_k^* would raise combinatorial issues. However, thanks to the repulsive nature of DPPs, these conditional probabilities decrease for inclusion.

Proposition 4.2. *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y,\ B \cap Y = \emptyset) \neq 0$, and let $k \neq l \in \overline{A \cup B}$. If $\mathbb{P}(A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) > 0$, then*

$$\mathbb{P}(\{k\} \subset Y | A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset).$$

If $\mathbb{P}(A \subset Y,\ (B \cup \{l\}) \cap Y = \emptyset) > 0$, then

$$\mathbb{P}(\{k\} \subset Y | A \subset Y,\ (B \cup \{l\}) \cap Y = \emptyset) \geq \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset),$$

Consequently, for all $k \in \mathcal{Y}$, if $y_{1:k-1} \leq z_{1:k-1}$ (where $\leq$ stands for the inclusion partial order) are two states for $Y_{1:k-1}$, then

$$\mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}) \geq \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}).$$

In particular, $\forall k \in \{1, \dots, N\}$, if $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ then

$$q_k^* = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) = K(k,k) + K_{k \times \{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\} \times k}.$$

Proof. Recall that by Proposition 3.1, $P(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset) = H^B(k,k) - H^B_{\{k\} \times A}(H^B_A)^{-1} H^B_{A \times \{k\}}$. Let $l \notin A \cup B \cup \{k\}$. Consider L^B the Cholesky decomposition of the matrix H^B obtained with the following ordering the coefficients: A, l, the remaining coefficients of $\mathcal{Y} \setminus (A \cup \{l\})$. Then, the restriction L^B_A is the Cholesky decomposition (of the reordered) H^B_A and thus

$$H^B_{\{k\} \times A}(H^B_A)^{-1} H^B_{A \times \{k\}} = H^B_{\{k\} \times A}(L^B_A (L^B_A)^*)^{-1} H^B_{A \times \{k\}} = \|(L^B_A)^{-1} H^B_{A \times \{k\}}\|_2^2.$$

Similarly,

$$H^B_{\{k\} \times A \cup \{l\}}(H^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}} = \|(L^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}}\|_2^2.$$

Now remark that solving the triangular system with $b = (L^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}}$ amounts solving the triangular system with $(L^B_A)^{-1} H^B_{A \times \{k\}}$ and an additional line at the bottom. Hence, one has $\|b\|_2^2 \geq \|(L^B_A)^{-1} H^B_{A \times \{k\}}\|_2^2$. Consequently, provided that $\mathbb{P}(A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) > 0$,

$$\mathbb{P}(\{k\} \subset Y | A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset).$$

The second inequality is obtained by complementarity in applying the above inequality to the DPP $\overline{Y}$ with $B \cup \{l\} \subset \overline{Y}$ and $A \cap \overline{Y} = \emptyset$. □

As a consequence, an admissible choice for the distribution of the Bernoulli process is

$$q_k = \begin{cases} \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) \\ \qquad \text{if } \mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0, \\ 1 \qquad \text{otherwise.} \end{cases} \quad (5)$$

Remark that if for some index k, $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ is not satisfied, then for all the subsequent indexes $l \geq k$, $q_l = 1$, that is the Bernoulli process becomes degenerate and contains all the points after k. In the remaining of this section X will denote a Bernoulli process with probabilities (q_k) given by (5).

As discussed in the previous section, in addition to being easily simulated, one would like the cardinal of X to be close to the one of Y. The next proposition shows that this is verified if all the eigenvalues of K are strictly less than 1.

Proposition 4.3 ($|X|$ is proportional to $|Y|$)**.** *Suppose that $P(Y = \emptyset) = \det(I - K) > 0$ and denote by $\lambda_{\max}(K) \in [0,1)$ the maximal eigenvalue of K. Then,*

$$\mathbb{E}(|X|) \leq \left(1 + \frac{\lambda_{\max}(K)}{2\,(1 - \lambda_{\max}(K))}\right) \mathbb{E}(|Y|). \quad (6)$$

Proof. By Proposition 3.1, $q_k = K(k,k) + K_{\{k\}\times\{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1}K_{\{1:k-1\}\times\{k\}}$. Since

$$\|((I-K)_{\{1:k-1\}})^{-1}\|_{\mathcal{M}_{k-1}(\mathbb{C})} = \frac{1}{1-\lambda_{\max}(K_{\{1:k-1\}})}$$

and $\lambda_{\max}(K_{\{1:k-1\}}) \leq \lambda_{\max}(K)$ one has,

$$K_{\{k\}\times\{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1}K_{\{1:k-1\}\times\{k\}} \leq \frac{1}{1-\lambda_{\max}(K)}\|K_{\{1:k-1\}\times\{k\}}\|_2^2.$$

Summing all these inequalities gives

$$\mathbb{E}(|X|) \leq \mathrm{Tr}(K) + \frac{1}{1-\lambda_{\max}(K)}\sum_{k=1}^{N}\|K_{\{1:k-1\}\times\{k\}}\|_2^2.$$

The last term is the Frobenius norm of the upper triangular part of K, hence in can be bounded by $\frac{1}{2}\|K\|_F^2 = \frac{1}{2}\sum_{j=1}^N \lambda_j(K)^2$. Since $\lambda_j(K)^2 \leq \lambda_j(K)\lambda_{\max}(K)$, $\sum_{j=1}^N \lambda_j(K)^2 \leq \lambda_{\max}(K)\,\mathrm{Tr}(K) = \lambda_{\max}(K)\mathbb{E}(|Y|)$. □

Algorithm 3 sampling of a DPP by sequential thinning of an adapted Bernoulli process

1. Compute sequentially the probabilities $\mathbb{P}(X_k = 1) = q_k$ of the Bernoulli process X:

- Compute the Cholesky decomposition L of the matrix $I-K$.
- For $k = 1$ to N:
  - If $q_{k-1} < 1$ (with the convention $q_0 = 0$),

    $$q_k = K(k,k) + \|L^{-1}_{\{1,\dots,k-1\}}K_{\{1,\dots,k-1\}\times\{k\}}\|_2^2$$

  - Else, $q_k = 1$.

2. Draw the Bernoulli process X. Let $m = |X|$ and $k_1 < k_2 < \cdots < k_m$ be the points of X.

3. Apply the sequential thinning to the points of X:

- Attempt to add sequentially each point of X to Y:
  Initialize $A \leftarrow \emptyset$ and $B \leftarrow \{1,\dots,k_1-1\}$
  For $j = 1$ to m
  - If $j > 1$, $B \leftarrow B \cup \{k_{j-1}+1,\dots,k_j-1\}$
  - Compute the conditional probability $p_{k_j} = \mathbb{P}(\{k_j\} \subset Y | A \subset Y,\ B \cap Y = \emptyset)$ (see Algorithm 2 for details).
  - Add k_j to A with probability $\frac{p_{k_j}}{q_{k_j}}$ or to B otherwise.
- Return A.

Algorithm 3 presents the different steps of our sequential thinning algorithm to sample a DPP of kernel K.

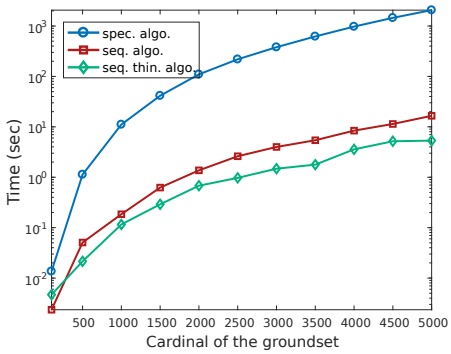

Figure 1: Running times of the 3 studied algorithms in function of the size of the groundset. The number of sampled points is constant (20).

Step 1 of Algorithm 3 is a preprocess that must be done only once for a given matrix K. Step 2 is trivial and fast. The critical point is to sequentially compute the conditional probabilities $p_k = \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset)$ for each point of X. Recall that in Algorithm 2 we use a Cholesky decomposition of the matrix $(I-K)_B$ which is updated by adding a line each time a point is added in B. Here, the inverse of the matrix $(I-K)_B$ is only needed when visiting a point $k \in X$, so one updates the Cholesky decomposition by block, where the new block corresponds to all indices added to B in one iteration.

Now, we will show experimentally that this algorithm enables to speed up the sampling of DPPs for some applications.

5 Experiments

5.1 Global and Detailed Running Times

In following experiments, we ran the algorithms on a laptop HP Intel(R) Core(TM) i7-6600U CPU and the software Matlab. First, let us compare the sequential thinning algorithm 3 presented here with the two main sampling algorithms: the classic spectral algorithm 1 and the "naive" sequential algorithm 2. Figure 1 presents the running times of the three algorithms as a function of the total number of points of the groundset. Here, we have chosen a common DPP kernel, a discrete adaptation of the Ginibre kernel. The expected cardinal $\mathbb{E}(|Y|)$ is constant, equal to 20. As foreseen, the sequential algorithm is far slower than the two others. Whatever the chosen kernel and the expected cardinal of the DPP, this algorithm is not competitive. Note that the sequential thinning

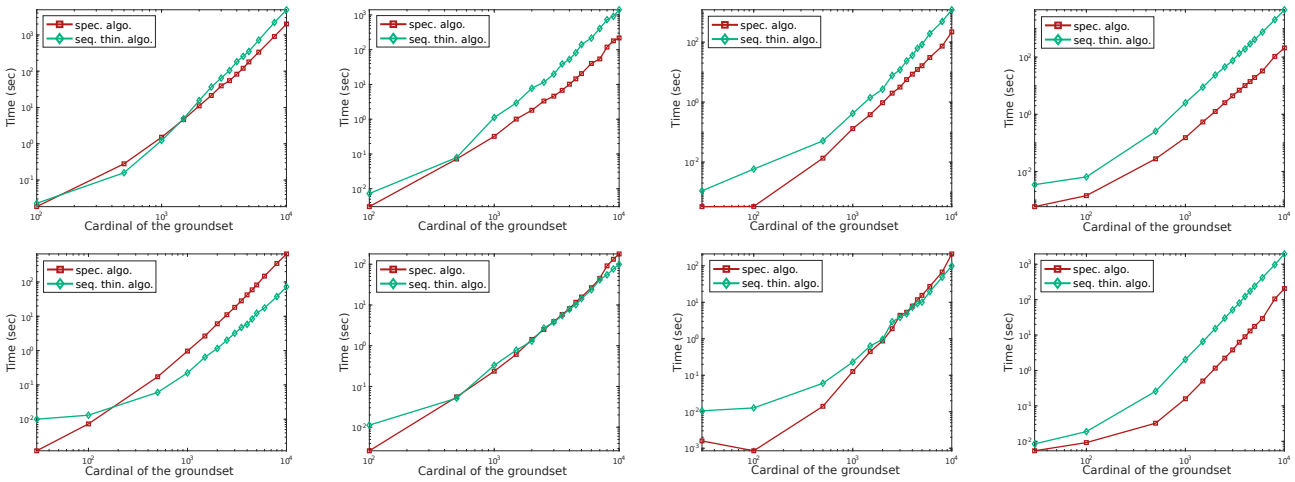

Figure 2: Running times in logscale of the spectral and the sequential thinning algorithms as a function of the size of the groundset $|\mathcal{Y}|$, using 3 "classic" DPP kernels. From left to right: a random kernel, a Ginibre kernel, a kernel based on the similarity between patches of an image and a projection kernel. On the first row, the expectation of the number of points is set to 4% of the $|\mathcal{Y}|$ and on the second row, $\mathbb{E}(|Y|)$ is constant, equal to 20.

algorithm uses this sequential method after sampling the particular Bernoulli process. But we will see that this first dominating step can be very efficient and lead to a fast algorithm.

Now, we can compare the spectral and the sequential thinning algorithms. We present in Figure 2 the running times of these algorithms as a function of the size of $|\mathcal{Y}|$ in several situations. The first row shows the running times when the expectation of the number of sampled point $\mathbb{E}(|Y|)$ is equal to 4% of the size of $\mathcal{Y}$: it increases as the total number of points increases. In this case, we can see that whatever the chosen kernel, the spectral algorithm is faster. On the second row, as $|\mathcal{Y}|$ grows, $\mathbb{E}(|Y|)$ is fixed to 20. Note that in this case the sequential thinning algorithm is competitive with the spectral algorithm. It is faster when using a random kernel and seems equivalent when using a Ginibre kernel or our third example of DPP kernel gathering similarities between the patches of an image.

The projection kernel (when the eigenvalues of K are either 0 or 1) is, as expected, a complicated case: Figure 2 (bottom, right) shows that our algorithm is not competitive when using this kernel. The main reason is that the cardinal of the dominating Bernoulli process X can be very large. In this case, the bound (6) isn't valid (and even tends to infinity) as $\lambda_{\max} = 1$, and we can quickly reach the degenerated case when, after some index k, all the probability $q_l, l \geq k$, are equal to 1. Then the second part of the sequential thinning algorithm -the sequential sampling part- is done on a larger set which significantly increases the running time of our algorithm.

Table 1: Mean of the detailed running times of the sequential, spectral and sequential thinning algorithms with $|\mathcal{Y}| \in [100, 10000]$ and a Ginibre kernel

Algorithms	Steps	Expected cardinal 4% of $\lvert\mathcal{Y}\rvert$	Expected cardinal 20
Sequential	Matrix inversion	85, 31%	85, 38%
	Choleski computation	13, 37%	12, 82%
Spectral	Eigendecomposition	91, 49%	98, 82%
	Sequential sampling	8, 15%	0, 783%
Sequential thinning	Preprocess to define q	5, 10%	18, 40%
	Sequential sampling	94, 86%	81, 55%

Thus, when the application needs to sample a large number of points, our algorithm is not competitive. Yet, when the size of the groundset is high and the number of points to sample is bounded, the sequential sampling is very competitive. Observe Figure 3: except when using a projection kernel, the sequential thinning algorithm is faster under a certain expected number of points -which depends on the kernel. For instance, when the kernel is randomly defined and the range of desired points to sample is below 200, it is relevant to use this algorithm. These requirements are quite common, for instance to summarize a text, to work only with representing points of clusters or to denoise an image with a patch-based method. See the next subsection for examples of these applications.

Table 1 presents the individual weight of the main steps of the three algorithms. Concerning the sequential algorithm, logically, the matrix inversion is the heaviest part taking 85.31% of the global running time. These proportions remain the same when the number of datapoints grows. The principal operation of the

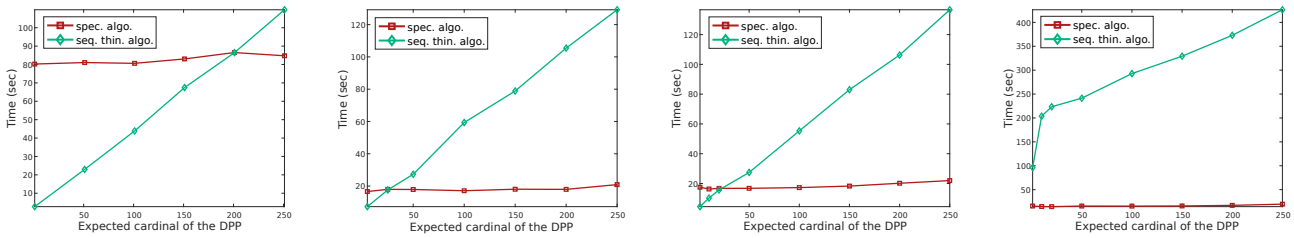

Figure 3: Running times of the spectral and sequential thinning algorithms in function of the expectation of the cardinal of the process. From left to right, using a random kernel, the Ginibre kernel, a kernel based on the similarity between patches of an image and a projection kernel. The size of the groundset is fixed to 5000 in all examples.

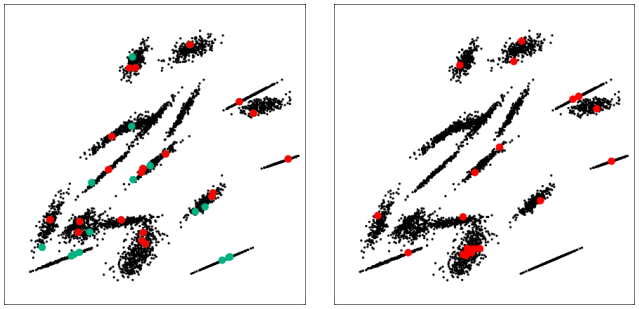

Figure 4: On the left, sampling of a DPP thanks to the sequential thinning algorithm (in green, the Bernoulli process of the first step, in red, the obtained DPP included in the Bernoulli process). On the right, an iid uniform point process. Samplings are on a dataset of 5000 points, with an expected cardinal equal to 20.

spectral algorithm is by far the eigendecomposition of the matrix K, counting for at least 90% of the global running time, when the expectation of the number of points to sample evolves with the size of $\mathcal{Y}$. Finally, the sequential sampling is the heaviest step of the sequential thinning algorithm. We have already mentionned that the thinning is very fast and that it produces a point process with a cardinal as close as possible to the final DPP. When the expected cardinal is low, the number of selected points by the thinning process is low too so the sequential sampling part stay bounded (81.55% when the expected cardinal $\mathbb{E}(|Y|)$ is constant). On the contrary, when $\mathbb{E}(|Y|)$ grows, the number of points selected by the dominated process rises as well so the running time of this step is growing (with a mean of 94.86%). As seen before, the global running time of the sequential thinning algorithm really depends on how good the domination is.

5.2 Application

We have seen that the sequential thinning algorithm is particularly efficient to sample a bounded number of points from a large groundset. The search for a subset of representative points from a large number of data gathered in clusters is a typical application of this framework. For instance, the initialization of a k-means algorithm, where you are looking for a first guess for the centroids of the clusters, needs to be fast and accurate enough because the results depends on it. Let suppose we have 5000 points gathered in 20 clusters, as shown in Figure 4. Then, it possible to generate a DPP kernel penalizing low distances between points with an expected cardinal equal to 20. Thanks to this DPP model, we can sample initialization points in a more repulsive way than by using an iid uniform point process with the same parameter $p = 20/5000$ for all points. This improves the accuracy of the k-means algorithm, given by the distorsion measure which is the mean square distance of the points to the centroid of their final cluster. After the initialization by the DPP of Figure 4, the final distorsion is 24.5% lower than after a random initialization. An initialization using DPPs enables to be closer to the real solution and here, using our algorithm doesn't cost too much.

6 Discussion

In this paper, we proposed a new sampling algorithm adapted to general determinantal point processs, which doesn't use the spectral decomposition of the kernel and is exact. It is an algorithm in two phases. The first one samples a Bernoulli process whose distribution is adapted to that of the targeted DPP. It is a fast and efficient step that reduces the initial number of points of the groundset. We know that if $(I - K)$ is invertible, the expectation of the cardinal of the Bernoulli process is proportional to the expectation of the cardinal of the DPP. Moreover, even if this first sampling procedure

may become degenerate (as soon as there is k such that $\mathbb{P}(X_k = 1) = 1$, all the subsequent points are selected), one can always change the visiting order to decrease the expected cardinal of the Bernoulli process. In practice, even when $(I - K)$ isn't invertible, the cardinal of the Bernoulli process remains low in comparison with the cardinal of the groundset. The second phase is a sequential sampling from the points selected in the first step. This algorithm is made possible thanks to the explicit formulations of the general marginals and the pointwise conditional probabilities of any DPP from its kernel K. Using updated Cholesky decompositions to compute the conditional probabilities, we fastened the sampling, even if its running time increases significantly with the size of its starting set of points.

In terms of running times, we have seen that for most examples, this algorithm is competitive with the spectral algorithm, and even sometimes faster, in particular when the size of the groundset is high and the expected cardinal of the DPP is reasonnable. This framework is common in machine learning issues. We took an example of application for which the use of the sequential thinning algorithm is relevant. Indeed, DPPs offer nice results to subsample a dataset, initialize a segmentation algorithm or summarize an image.

References

[1] R. H. Affandi, A. Kulesza, E. B. Fox, and Ben Taskar. Nystrom approximation for large-scale determinantal processes. In *AISTATS*, volume 31 of *JMLR Workshop and Conference Proceedings*, pages 85–98. JMLR.org, 2013.

[2] N. Anari, S. Ov. Gharan, and A. Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *COLT*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 103–115. JMLR.org, 2016.

[3] C. Dupuy and F. Bach. Learning determinantal point processes in sublinear time. Accepted to AISTATS 2018, Oct. 2016.

[4] M. Gartrell, U. Paquet, and N. Koenigstein. Low-rank factorization of determinantal point processes for recommendation. *ArXiv e-prints*, Feb. 2016.

[5] G. Gautier, R. Bardenet, and M. Valko. Zonotope hit-and-run for efficient sampling from projection DPPs. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR, Aug. 2017.

[6] J. Gillenwater, A. Kulesza, and B. Taskar. Discovering diverse and salient threads in document collections. In *EMNLP-CoNLL*, pages 710–720. ACL, 2012.

[7] J. Ginibre. Statistical ensembles of complex: Quaternion, and real matrices. *Journal of Mathematical Physics*, Vol: 6, Mar 1965.

[8] J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probability Surveys*, pages 206–229, 2006.

[9] B. Kang. Fast determinantal point process sampling with application to clustering. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2319–2327. 2013.

[10] T. Kathuria, A. Deshpande, and P. Kohli. Batched gaussian process bandit optimization via determinantal point processes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4206–4214. Curran Associates, Inc., 2016.

[11] A. Kulesza and B. Taskar. Structured determinantal point processes. In *NIPS*, pages 1171–1179. Curran Associates, Inc., 2010.

[12] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012.

[13] A. Kulesza and B. Taskar. Learning determinantal point processes. *CoRR*, abs/1202.3738, 2012.

[14] F. Lavancier, J. Møller, and E. Rubak. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4):853–877, 2015.

[15] C. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1328–1337, Cadiz, Spain, 09–11 May 2016. PMLR.

[16] C. Li, S. Sra, and S. Jegelka. Fast mixing markov chains for strongly rayleigh measures, dpps, and

constrained sampling. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4188–4196. Curran Associates, Inc., 2016.

[17] J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. *J. Algorithms*, 27(2):170–217, 1998.

[18] T. Rolski and R. Szekli. Stochastic ordering and thinning of point processes. *Stochastic Processes and their Applications*, 37(2):299–312, 1991.

[19] A. Scardicchio, C. E. Zachary, and S. Torquato. Statistical properties of determinantal point processes in high dimensional euclidean spaces. *Phys. Rev. E*, 79(4), 2009.

[20] C. Zhang, H. Kjellström, and S. Mandt. Balanced mini-batch sampling for SGD using determinantal point processes. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, Aug. 2017.